# Halftone Primitive Watermarking and Related Applications

## Technical Field

This patent application claims the benefit of application 60/263,987, filed January 24, 2001, which is hereby incorporated by reference.

## Background and Summary

Digital watermarking is a process for modifying physical or electronic media to embed a machine-readable code into the media. The media may be modified such that the embedded code is imperceptible or nearly imperceptible to the user, yet may be detected through an automated detection process. Most commonly, digital watermarking is applied to media signals such as images, audio signals, and video signals. However, it may also be applied to other types of media objects, including documents (e.g., through line, word or character shifting), software, multi-dimensional graphics models, and surface textures of objects.

Digital watermarking systems typically have two primary components: an encoder that embeds the watermark in a host media signal, and a decoder that detects and reads the embedded watermark from a signal suspected of containing a watermark (a suspect signal). The encoder embeds a watermark by altering the host media signal. The reading component analyzes a suspect signal to detect whether a watermark is present. In applications where the watermark encodes information, the reader extracts this information from the detected watermark.

Several particular watermarking techniques have been developed. The reader is presumed to be familiar with the literature in this field. Particular techniques for embedding and detecting imperceptible watermarks in media signals are detailed in the assignee's co-pending U.S. Application Serial No. 09/503,881 and U.S. Patent 5,862,260, which are hereby incorporated by reference.

The invention provides a method for embedding a digital watermark into a halftone image comprised of clusters of halftone dots, which are referred to as geometric

primitives. In the literature on clustered-dot halftone printing, these geometric primitives are sometimes referred to as halftone dots. For color halftone images, primitives of different colors (CMYK or RGB) are combined at a pixel location to form a clustered dot having a desired color. In this document, the cluster that forms a primitive is comprised of microdots referred to as halftone dots or halftone pixel dots. The method embeds a watermark image signal comprising an array of multilevel per pixel delta values by increasing or decreasing the size of a corresponding cluster of halftone dots. In particular, each delta value represents a change in luminance in one implementation, but this delta value could correspond to other color planes of a monotone or color halftone image. The watermark embedder subtly changes the halftone primitives by toggling boundary pixels on or off in a manner that grows or shrinks the primitive.

A watermark decoder detects and reads a message in the watermarked image after it is scanned from a printed object, such as a newspaper, by detecting the changes in luminance (or other color plane) in which the watermark is embedded.

Further features will become apparent with reference to the following detailed description.



## Detailed Description

The following description details methods for watermarking halftone images and related applications. A halftone image is represented as a two-dimensional array of image elements (pixels), each having a one bit binary value of one or zero. In printing or displaying such an image, each binary value corresponds to the presence or absence of a signal, such as a black or white dot. In color halftone images, each pixel location has a vector of 1 bit binary values, where the elements in the vector correspond to color components like RGB, CMYK, etc. Many digital images are represented as multilevel per pixel images, where each pixel has one or more multilevel color values (one for grayscale, and three or four typically for color). Halftone processes convert multilevel per pixel images at a source resolution to halftone images at a higher target resolution by subdividing each 2D region representing a location of a multilevel pixel image in a

rectangular array into a 2D array of halftone pixel locations and selectively setting bits of the halftone locations to one or zero to represent the value of the multilevel pixel.

There are a variety of halftone processes. One technique used in newsprint, for example, is to represent multilevel pixel values as 2D geometric primitives, comprised of contiguous clusters of halftone pixel dots (e.g., black dots). These clusters are in the shape of solid color shapes, such as rectangles or polygons that appear circular from a distance. In a grayscale image, shades of gray from light (high luminance) to dark (low luminance) are represented as progressively larger solid primitives that occupy progressively larger areas of a given multilevel pixel location. These geometric primitives are typically centered within a rectangle representing the multilevel pixel location.

In one method detailed below, a watermark encoder modulates the size of halftone primitives to encode a watermark signal. This method operates on a halftone image that is in the form of a collection of geometric primitives (e.g., clusters of black halftone dots). Another input is a watermark signal, which is represented as a multilevel per pixel image at some lower resolution than the resolution of the halftone dots in the halftone image. Elements in the watermark signal represent changes in luminance of the halftone image, and each correspond to a halftone primitive comprising a cluster of halftone dots. To embed the watermark in the halftone image, the watermark encoder toggles halftone dot values at the outer boundary of a primitive to effect the desired change in luminance of the corresponding multilevel pixel value in the watermark signal. It repeats this process for each element in the watermark signal.

The effect of the watermark on a grayscale image represented as clusters of black or white halftone dots is to subtly change the luminance of the halftone image. The degree of imperceptibility depends on the magnitude of the watermark signal. The magnitude of the watermark signal is scaled, either by a user supplied global gain applied evenly across the image or regions of the image, or by an automated process that varies the gain applied to regions of the image based on a human visual system modeling of the

image. One way to model the image is to compute local contrast and increase the gain in areas of high contrast.

Once watermarked, a conventional printing process prints the watermarked halftone image by applying ink dots only to areas where halftone dots are present in the digital representation of the halftone image.

To read the watermark, a digital image scanning process captures a digital image at sufficient resolution (at least as great as the resolution of the watermark signal). A watermark decoder converts the image into luminance, detects the presence and orientation of the watermark signal, and then decodes a binary message embedded in the watermark signal.

There are a number of ways to compute the watermark signal, and to decode it from a digital image scan of a printed image. Also, of course, the method described above applies to halftone images that remain in the digital domain, e.g., are not printed. Some examples of computing the watermark signal and decoding it are provided in co-pending U.S. Application Serial No. 09/503,881 and U.S. Patent 5,862,260, which are incorporated by reference above.

One approach for creating the watermark signal is to take a desired binary message comprising a sequence of binary values, append known control bits, error correction encode the entire sequence, spread each bit in the sequence over a pseudorandom number (e.g., a random binary sequence), and map the resulting sequence elements to pixel locations in a 2D image. The sequence may be represented as a binary antipodal sequence of 1 and –1, where 1 corresponds to a binary 1 and –1 corresponds to a binary zero. To give the elements a multilevel value, the watermark embedder scales the antipodal values by an array of gain values, where elements in the array correspond to elements in the watermark signal. Additionally, the watermark signal may be combined with a calibration signal, such as a signal represented as a collection of delta values in the Fourier magnitude domain, with each delta value having pseudorandom phase.

To detect the watermark, a watermark detector correlates the calibration signal with the scanned version of the watermarked image to determine the original rotation,

scale and origin of the watermark signal. Then, a message reader decodes the watermark message by comparing pixel values with predicted values (e.g., a local average of a neighboring pixels). If a pixel value is greater than a predicted value, it is mapped to a binary one; otherwise it is mapped to a zero. The resulting values are applied to an inverse spreading operation and then an inverse error correction coding operation. The known bits are then used to validate the message, including validating the message by checking error detection bits in an error detection process.

The following discussion illustrates an implementation in further detail. In this implementation, the primitives are rectangles and clusters of dots that approximate circles. The embedder modulates halftone primitives as follows:

1. To make primitives bigger, weight edge halftone pixels by weighting pixels in a manner that determines the priority order in which they are turned on (changed from white to black). This order is calculated to preserve the shape of the primitive. For example, the corners are given a greater weight.

2. To make primitives smaller, turn off edge pixels using a similar weighting to prioritize pixels to be changed.

3. The delta value for each watermark signal pixel indicates how many edge pixels to turn on to make a negative change in luminance, and how many to turn off to make a positive change in luminance.

4. The watermark signal is a tile of pseudorandom numbers, modulating corresponding halftone cluster sizes up or down. The tile is replicated in the host image.

5. This approach prioritizes changes along edges of clusters by making the change along one edge first, before moving to another edge. This makes the change less perceptible. In general, the priority scheme for changing edge pixels shrinks or grows a primitive uniformly only at the edge.

6. This approach allows the embedder to embed more signal strength for less visibility because it modulates the halftone cluster primitive that the image is being drawn with.

7. The weighting priority for adding or removing dots along a boundary may be generalized for all shapes or adapted to be optimized for different shapes. When optimized for different shapes, the embedder selects the weighting scheme applicable to the cluster shape being modified.

8. The weighting can be calculated with a convolution filter that establishes a sequence for turning on halftone pixels in a uniform manner to grow a halftone cluster primitive, and similarly turning off pixels on a boundary in a uniform manner to shrink a halftone cluster primitive.

9. One implementation examines horizontally and vertically adjacent pixels on cluster edges, and sums nearest neighbors (vertically adjacent and horizontally adjacent) of opposite color, distinguishing between cases for growing or shrinking a cluster. In the case where luminance is to be increased, this method prioritizes, for example, black edge pixels surrounded by more white pixels, and turns these black pixels to white first. In the case where luminance is to be decreased, this method prioritizes, for example, white pixels surrounded by more black pixels, and turns these white pixels to black first. This method prioritizes runs of horizontally and vertically adjacent edge pixels such that runs of such edge pixels are expanded outward or shrunk inward together. If the delta number of the watermark signal pixel corresponding to the cluster being examined requires that more than one edge be changed, the method iterates through the modified primitive in more than one pass. With each iteration, the method prioritizes edge pixels of the primitive cluster from the previous pass and toggles edge pixels of higher priority until the total number of edge pixel changes corresponding to the delta value is made.

## *Combined Robust and Fragile Watermark for Authentication of Printed Objects*

This section describes a method for halftone watermarking to determine whether a printed image has been tampered with. In this method, the watermark embedder embeds a robust watermark signal such as the one described above that is robust to printing, scanning and geometric distortion. For each pixel in the watermark signal, the embedder adds or subtracts halftone dots in corresponding block of halftone dots to achieve a

desired change in luminance. This method uses a random halftone dither pattern of dots to perform the halftone process. For each watermark signal element, the method adds or subtracts halftone dots to achieve the desired luminance change. However, the halftone patterns of the watermarked halftone image are selected to achieve the desired change in luminance and to have a predetermined dither pattern. The dither pattern is known, and may be pseudorandom.

In the watermark decoder, a detector uses the approach described above to detect the presence and orientation of the watermark. It then extracts the message carried in the robust watermark. The message may carry a key indicating the location and types of dither patterns used to create the watermarked image. Whether provided in the robust message or not, the decoder has a key that defines the dither patterns. The decoder then examines the realigned data to see whether the dither patterns are present. If they are within some tolerance or threshold settings, then the printed image is deemed authentic. If not, then it is deemed not authentic. The dither patterns used for authentication may be set within the printer of authentic objects.



### *Concluding Remarks*

Having described and illustrated the principles of the technology with reference to specific implementations, it will be recognized that the technology can be implemented in many other, different, forms. To provide a comprehensive disclosure without unduly lengthening the specification, applicant incorporates by reference the patents and patent applications referenced above.

The methods, processes, and systems described above may be implemented in hardware, software or a combination of hardware and software. For example, the auxiliary data encoding processes may be implemented in a programmable computer or a special purpose digital circuit. Similarly, auxiliary data decoding may be implemented in software, firmware, hardware, or combinations of software, firmware and hardware. The methods and processes described above may be implemented in programs executed from a system's memory (a computer readable medium, such as an electronic, optical or magnetic storage device).

The particular combinations of elements and features in the above-detailed embodiments are exemplary only; the interchanging and substitution of these teachings with other teachings in this and the incorporated-by-reference patents/applications are also contemplated.